Development of Routing Algorithms to Optimize Freight Delivery Trips

Authors

Surendra Reddy K, Gitakrishnan Ramadurai

Report I April 2020

Centre of Excellence in Urban Transport

Indian Institute of Technology, Madras

Shakti Sustainable Energy Foundation seeks to facilitate India's transition to a sustainable energy future by aiding the design and implementation of policies in the areas of clean power, energy efficiency, sustainable urban transport, climate change mitigation and clean energy finance.

Disclaimer

The views/analysis expressed in this report/document do not necessarily reflect the views of Shakti Sustainable Energy Foundation. The Foundation also does not guarantee the accuracy of any data included in this publication nor does it accept any responsibility for the consequences of its use.

*** For private circulation only**

Acknowledgement

We express deep gratitude to our sponsor Shakti Sustainable Energy Foundation (SSEF) for their support in the "Implementation Roadmap for Sustainable Urban Freight Mobility in Chennai" project.

We are grateful to Madras Chamber of Commerce and Industry (MCCI) for all the support and encouragement to our project team. We sincerely thank Ms Saraswathi, Secretary General and her team for organizing the workshop and their guidance during the course of the project.

We sincerely thank the staff members of ITS lab and doctoral research scholars of the Department of Civil Engineering, for their constant help in conducting surveys and the execution of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Freight transport is a vital component of the country's economy. However, quality of life in an urban area is affected by freight transport's disproportionately higher adverse effects such as worsening air quality and congestion. Urban freight transport has a composition of 10-15% of urban road traffic yet contributes to 40% of the air pollution (BESTUFS, 2006). These adverse effects can be minimized by optimizing the routes taken by these freight vehicles with respect to fuel consumption or emissions.

Vehicle Routing Problem (VRP), an important problem for any distribution company, is a combinatorial optimization problem where the goal is to find the optimal set of routes for a fleet of vehicles to traverse and deliver to a given set of customers. Here, the optimal routes can be in terms of distance (Renaud et al., 1996), fuel consumption (Kara et al., 2007), travel time (Li et al., 2010) , or total cost (Bektas et al., 2011). Recent research in this area focuses on optimizing with respect to either total cost or fuel consumption to align with operator objectives. An empirical analysis by Sahin et al. (2009) showed that the fuel and driver cost alone contributes to 60% of the overall cost for logistics operators, indicating the importance of minimizing fuel consumption from the operator's perspective. A report by US Environmental Protection Agency (USEPA) (*U.S. Climate Action Report, 2014*) estimated 33 percent of CO2 emissions from fossil fuel combustion in 2011 was from the transportation sector, of which nearly 63 percent of the emissions were from road transport. Minimizing fuel has the twin benefit of meeting the operator interest as well as environmental interests.

A typical solution to a VRP with one depot and five customers is shown in figure 1. Here, the number of vehicles required to deliver all the five customers that leads to minimum cost is two. The first vehicle delivers to customers 1 and 2 in that order, and the second vehicle delivers to customers 3, 4, and 5 in the order 5-4-3. VRP can be solved by minimizing distance or fuel consumption. It is well-known that solutions are likely to be different in the two cases. From a sustainable urban freight transport perspective, minimizing fuel consumption is more desirable.

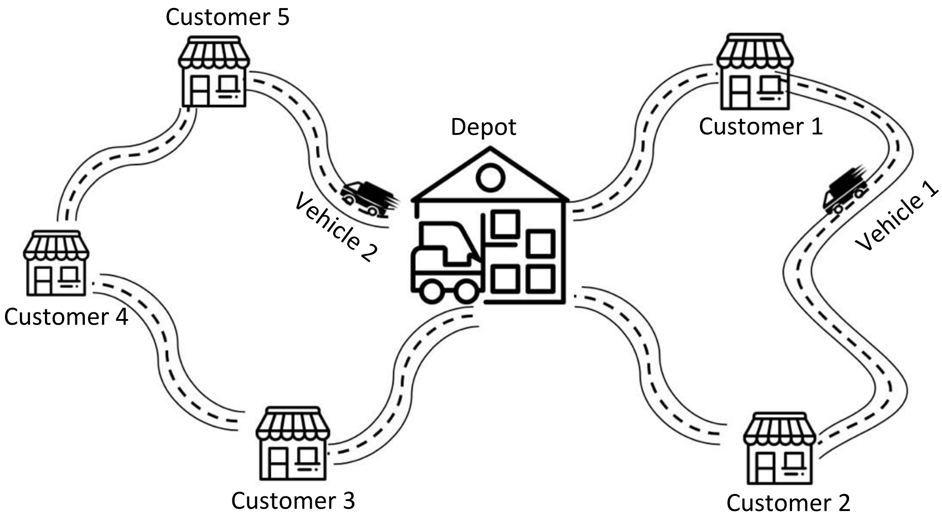

FIGURE 1 TYPICAL SOLUTION TO A VRP

FUEL CONSUMPTION ESTIMATION

Fuel consumption can be estimated using different models such as instantaneous fuel consumption model, four-mode elemental fuel consumption model, running speed fuel consumption model, Comprehensive Modal Emission Model (CMEM), and COmputer Programme to calculate Emissions from Road Transportation (COPERT) model. From the comparative analysis of these models (Demir et al., 2011) it was found that four-mode elemental fuel consumption model and CMEM closely represent the real-world scenario. However, these models require more vehicle specific data and second-by-second speed data. Driving cycle, a series of data points representing the second-by-second speed of a vehicle, can be used in fuel consumption estimation models. Driving cycles were generally derived from the movement of test vehicle under real traffic conditions; hence, they are also more representative of real-world driving pattern.

We have used Comprehensive Modal Emission Model (CMEM) (Barth et al., 2005, 2004) for fuel consumption estimation. This model was selected particularly because it takes into account all the important parameters such as speed, acceleration, load, and grade. Moreover, this model can be applied for heavy duty diesel vehicles as well as smaller pick-up trucks. CMEM consists of three modules, namely engine power module, engine speed module, and fuel rate module.

Engine Power (P) requirement is calculated using

$$P = \frac{(Ma + Mg\sin\theta + MgC_r\cos\theta + 0.5C_d\rho Av^2)v}{1000\epsilon} + P_{acc} \qquad (1)$$

where v is the speed (m/s), a is acceleration (m/s^2), M is the gross vehicle weight (kg), g is the gravitational constant (m/s^2), θ is the road grade angle in degrees, ρ is the air density (kg/m^3 ,typically 1.2041), A is the frontal surface area (m^2), C_d is the coefficient of aerodynamic drag, C_r the coefficient of rolling resistance, ε is the vehicle drive train efficiency (typically 0.4), P is the second-by-second engine power output (kW), and P_{acc} is the engine power demand associated with running losses of the engine.

Engine speed (N) is interpolated between idle rpm and governing rpm using the speed of vehicle at that moment and then Fuel Rate (FR) is calculated as follows

$$FR = \frac{\varphi(kNV + P\eta)}{U} \qquad (2)$$

where φ is fuel-to-air mass ratio (typically 0.0667), k is the engine friction factor (typically 0.2), V is the engine displacement (liters) and η is a measure of indicated efficiency for diesel engines (typically 0.45). U is lower heating value for the fuel (for diesel 44).

For ease of computing, equation 2 is simplified as follows

$$FR = \alpha + \beta M \qquad (3)$$

Where,

$\alpha = \dfrac{\varphi\left(kNV + \frac{0.5C_d\rho Av^3}{1000\eta\epsilon}\right)}{U}$ is the load independent part,

$\beta = \dfrac{\varphi(a + g\sin\theta + gC_r\cos\theta)v}{1000\epsilon\eta U}$ is the load dependent part, and

M is the load including curb weight of vehicle.

Equation 3 was used in preprocessing stage to calculate the matrix for α and β. Load data that is obtained while solving the model is used in combination with β and α in order to get the fuel consumption for traveling on a link. Travel time (s) is the time taken to travel the distance of each link with speeds varying as described in driving cycle. Here, we have used US EPA urban driving cycle developed for light duty vehicles (see figure 2).

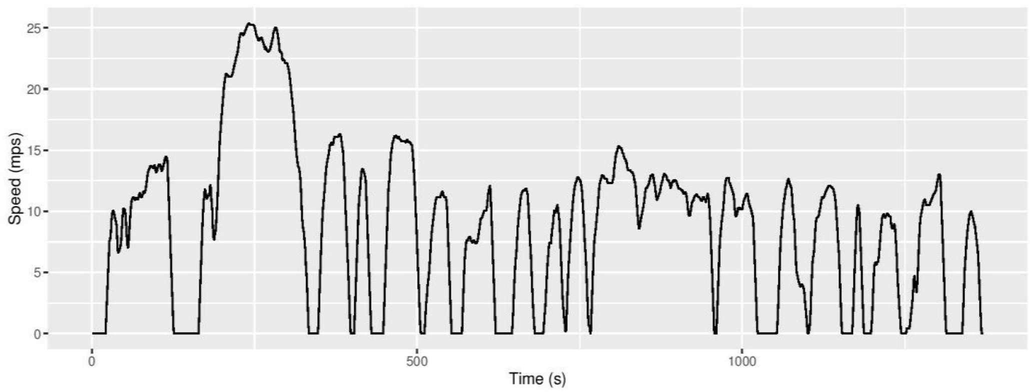

FIGURE 2 USEPA URBAN DRIVING CYCLE FOR LIGHT DUTY VEHICLES

ASSUMPTIONS

This model presented here is based on the following assumptions:

1. There is only one central depot from where all the deliveries happen for a given network.
2. All the vehicles used are from a homogeneous fleet.
3. An unlimited number of vehicles are available for deliveries.
4. All the links in the network have the same driving pattern of the driving cycle used.

The first two assumptions mentioned above are the main limitations of the model. However, the first assumption can be overcome by dividing the urban area into zones such that each zone has only one depot. The second assumption is not overly restrictive since most urban freight delivery happens using pick-ups and small trucks. The third assumption gives the best possible routes for the network. Since we are not imposing any time restrictions on deliveries, this assumption is not a major limitation. For example, if the solution is eight vehicles (routes) are required for optimal deliver and there are only 5 vehicles available, then 3 vehicles may be assigned two trips each while the remaining two vehicles each make a single trip. The fourth assumption can be relaxed if there is data available for each link in the network. Driving cycles derived from such data can be incorporated. Incorporating link-level data will result in accurate real-world representation, but it requires a large volume of data. Availability of GPS in most trucks could be a source of such large data.

Solution Algorithm

VRP is a NP-hard problem (Lenstra & Rinnooy Kan 1981). NP-hard problems are those that do not have algorithms that find solutions in polynomial time. Typically as the NP-hard problem size increases, the algorithm run times scale exponentially. Hence, we need to depend on heuristics to solve most real-world problems. Heuristics are mainly classified into two groups namely i) local search based, and ii) population based. From the literature it was found that local search-based heuristics perform better in solving VRP. There are numerous heuristic algorithms that can be grouped under local search and few of the popular heuristics in this group are Simulated Annealing (SA), Tabu Search, Large Neighborhood Search, Adaptive Large Neighborhood Search. Few of these heuristics perform better in terms of solution quality and others in terms of run time. It is a traditional tradeoff between runtime and solution quality. SA is one of the most widely used local search algorithm and its run time is low compared to other algorithms. However, its solution quality may not be the best. Here, we gave importance to run time since we are developing a near real-time web application, and hence selected SA.

SA (Kirkpatrick et al., 1983) is a local search technique for approximating the global optimum of a given function. It is analogs to the process of annealing used in metallurgy where a material is heated. Then the temperature is slowly decreased to reduce defects, thus minimizing the system energy. Similarly, SA improves the initial solution while slowly cooling the temperature. It accepts a new solution in the following two cases: i) if it is an improvement over the present, or ii) if it is worse, then with a probability determined by the Boltzmann function. The latter case will prevent the SA from getting trapped at a local optimum during the initial stages of the search.

SA needs an initial solution to start, then a new neighborhood of the present solution was obtained at each iteration using three techniques: namely, insertion, swap, and reversion (refer figure 3). These techniques were selected randomly to escape from any local optimum. In the insertion process, the algorithm selects two nodes randomly. Then the first node was inserted after the second node, whereas in the swap process, two nodes were chosen randomly and swapped. Finally, in reversion, two nodes were selected, and the path including all the nodes in between them was reversed.

The algorithm was allowed to find any solution (can be feasible or infeasible). However, we apply a heavy penalty for solutions that violate the constraints. The algorithm will stop if there was no improvement in the solution for 150 consecutive temperature decrements or when the temperature falls below a final temperature. The flow chart of the algorithm used is shown in figure 4.

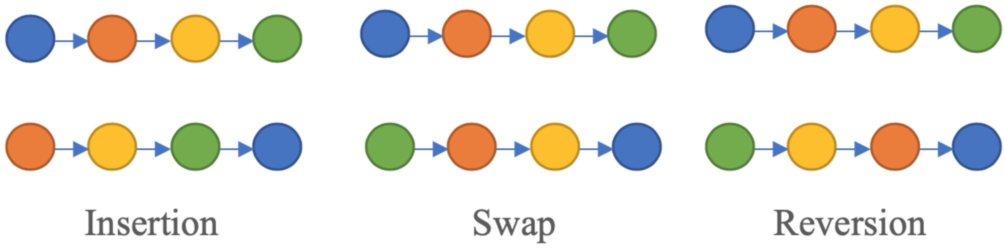

FIGURE 3 THREE TECHNIQUES USED IN SA

Performance of SA depends on the following parameters - iterations at a particular temperature (T_{iter}), initial temperature (T_i), final temperature (T_f), temperature reduction rate (α), and the Boltzmann constant (k). The parameters that gave the best results during the trial runs are as follows: T_{iter} = 800, T_i = 50, α = 0.95, T_f = 0.001 and k = 1/9.

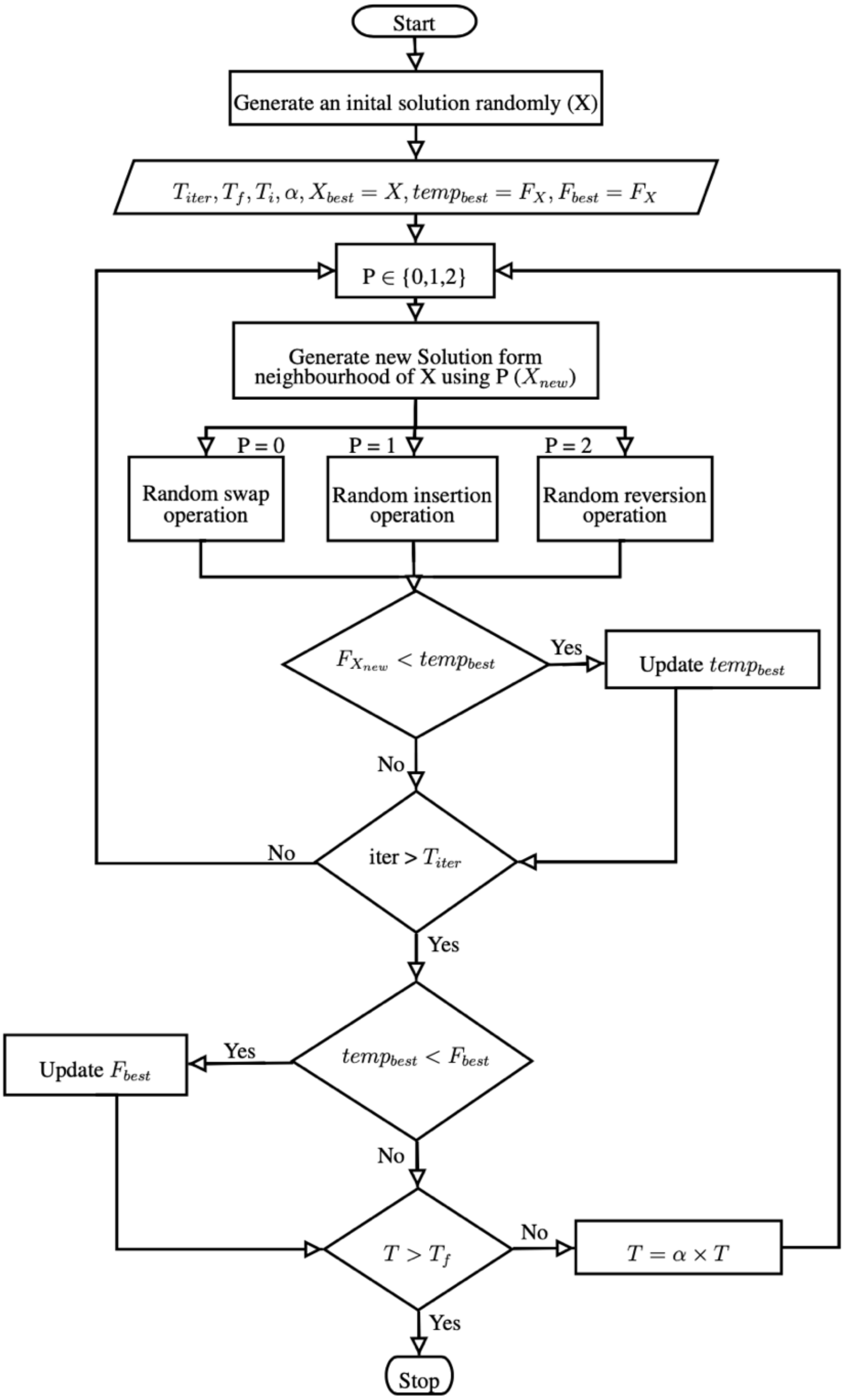

FIGURE 4 FLOW CHART OF SA ALGORITHM

WEB APPLICATION

A web application is developed using Flask for Python. This web application acts as an interface for the user and the algorithm. Here, the user can create a network of nodes consisting of customers and a depot by either using the search bar or with a click on the map. Figure 4 shows the user interface developed. Once the user selects a node, a pop-up will be displayed, showing the latitude and longitude of the location along with an option to choose the node type and input demand of the node. Figure 5 shows the pop-up displayed after selecting a point on the map. The user can repeat this process of selecting the node and providing input depending on the number of customers he wants to serve. After the user finishes the selection of customers and a depot on the map, then he can name it and provide the vehicle capacity details, which completes the data required to form the user. Figure 6 shows the location of all the nodes selected along with the name given to the network and capacity of the vehicles available at the user. Now, the user can click on the solve button that transfers all the input data to the SA algorithm. SA algorithm solves the problem and provides the optimal routes to serve the customers as output. The web application processes the output given by the SA algorithm and plots the routes on the map (figure 7).

The code and details to setup the server is available on https://github.com/surendrark/VRPonWEB.

Minimum system requirements required to run the server are as follows: Intel i3 @ 1.6 GHz CPU, 4 GB RAM, 128 GB HDD, Running Windows 7 or higher/ macOS 10.10 or higher.

To setup and use this server, basic knowledge of python and mysql is desirable, but not a strict requirement. Any organization/establishment that deals with deliveries of goods can use this application.

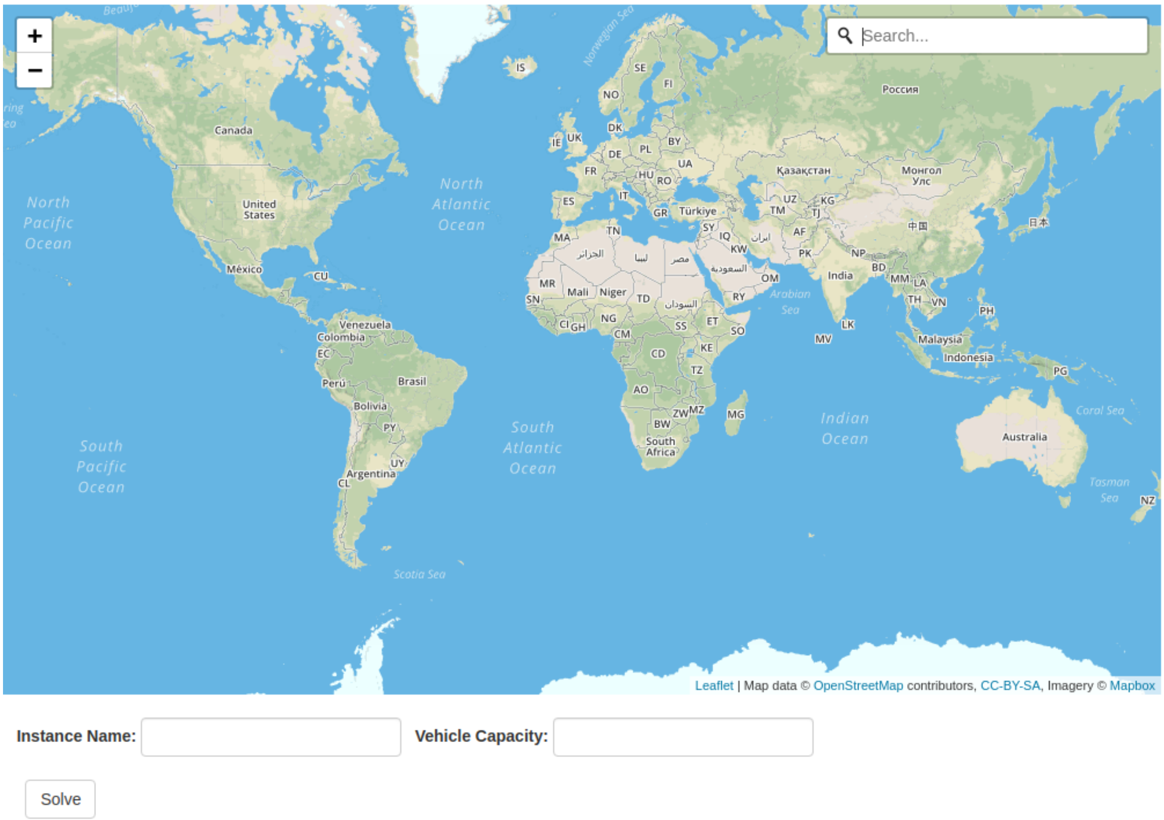

FIGURE 5 INITIAL INTERFACE OF THE WEB APPLICATION

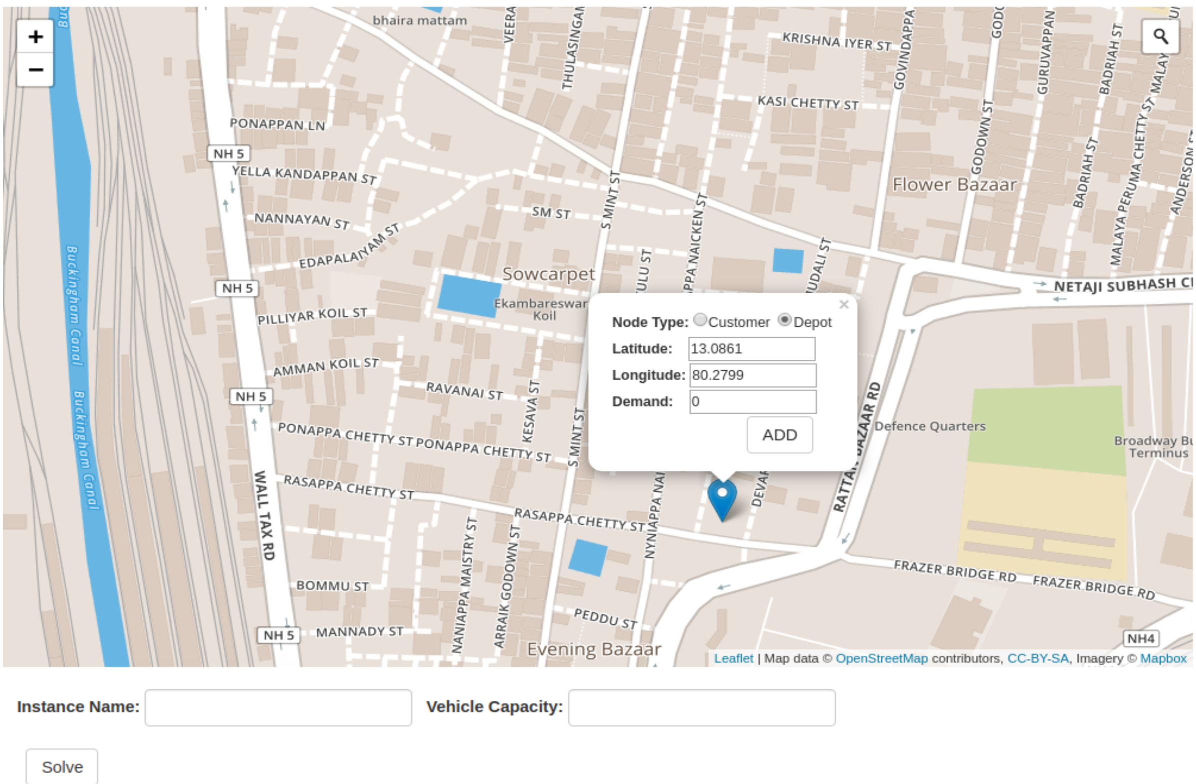

FIGURE 6 INPUT POP-UP FOR THE NODE SELECTED ON MAP

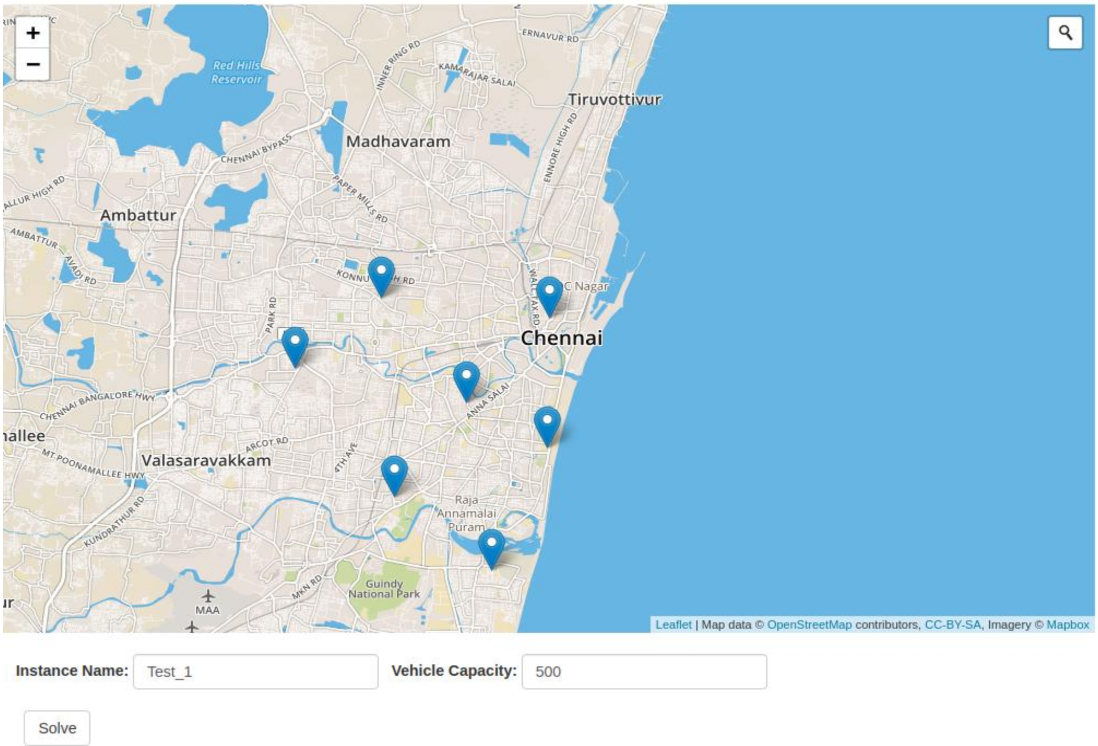

FIGURE 7 SCREEN SHOT OF THE INTERFACE WITH ALL NODES SELECTED

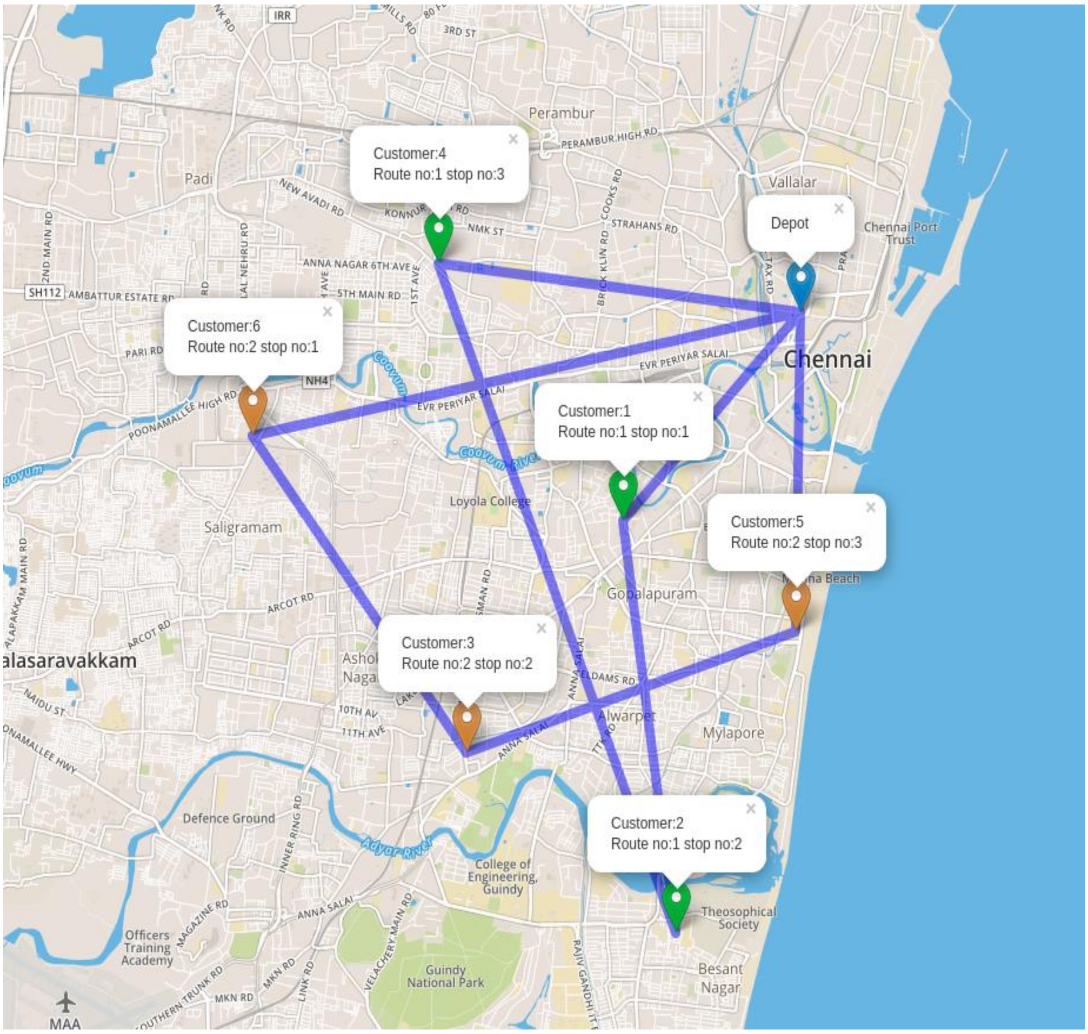

FIGURE 8 NEAR OPTIMAL ROUTES FOR THE GIVEN INPUT

ADVANTAGE

The advantage of using this tool is explained with the help of a sample problem.

Problem description

Consider a network with 20 customers and one depot. Our objective is to find the routes which lead to the lowest fuel cost. The location and demand information for the customers and depot are shown in table 1

TABLE 1 LOCATION AND DEMAND INFORMATION OF NODES IN THE NETWORK

Node	Demand	X co-ordinate	Y co-ordinate
Customer 1	110	151	264
Customer 2	70	159	261
Customer 3	80	130	254
Customer 4	140	128	252
Customer 5	210	163	247
Customer 6	40	146	246
Customer 7	80	161	242
Customer 8	10	142	239
Customer 9	50	163	236
Customer 10	60	148	232
Customer 11	130	156	217
Customer 12	130	129	214
Customer 13	30	146	208
Customer 14	90	164	208
Customer 15	210	141	206
Customer 16	100	147	193
Customer 17	90	164	193
Customer 18	250	129	189
Customer 19	180	155	185
Customer 20	70	139	182
Depot	NA	128	231

The algorithm presented here uses driving cycles to estimate fuel consumption accurately. To evaluate the effect of using the driving cycle, we have compared the results obtained by using driving cycle data with the results obtained using average speed data (see figure 8).

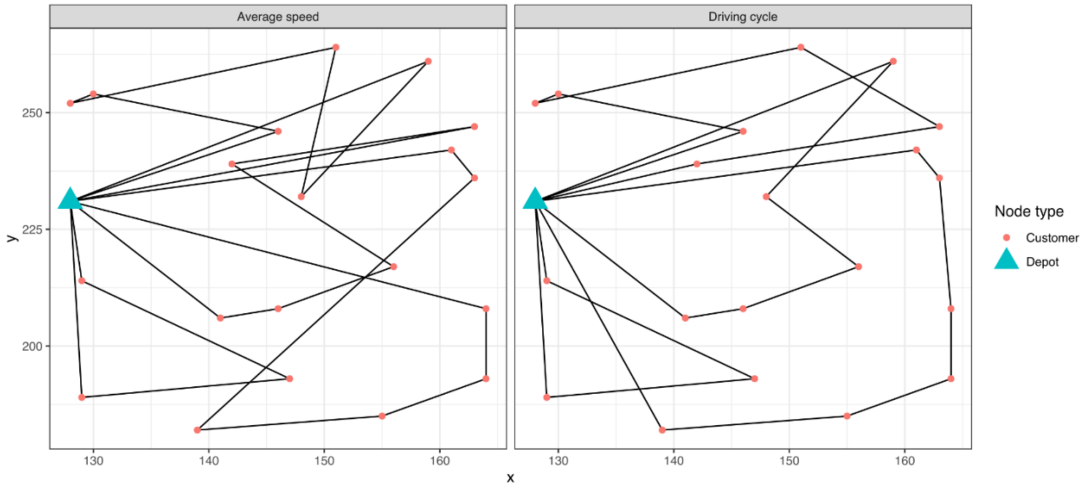

FIGURE 9 OPTIMAL ROUTES FOR AVERAGE SPEED CASE AND DRIVING CYCLE CASE

The optimal routes obtained by average speed case consumes **46.58** units of fuel and **601.82** units of distance traveled. In contrast, the driving cycle case consumes only **41.21** units of fuel and **535.47** units of distance traveled. This reduction is mainly because of the accurate fuel consumption estimation procedure used in the algorithm. Hence, taking the routes obtained by this algorithm may lead to savings in both fuel consumed and distance traveled.

SCOPE FOR FUTURE DEVELOPMENT

The web application can solve only single depot VRP with deliveries only. This can be extended to solve problems with multiple depot, simultaneous pickup and deliveries. The distances considered here are crow fly distance. But, we may include actual road distance with the help of the OSRM server that can also be set up locally. The code for which is available at the following location https://github.com/Project-OSRM/osrm-backend. The web application is developed for single user purpose only, which can be extended such that multiple users can access it at the same time. The SA algorithm used here may not be the best. However, we may incorporate any new algorithm with only a couple of changes. First, the input from web application needs to be formatted to suit the format of the algorithm, and second, the output from the algorithm needs to be formatted to suit the format of the web application. The web application is coded in a modular fashion that makes incorporation of the above-mentioned updates swift and easy.

REFERENCES

1. Barth, M., Scora, G., Younglove, T., 2004. Modal Emissions Model for Heavy-Duty Diesel Vehicles. Transp. Res. Rec. J. Transp. Res. Board 1880, 10–20.
2. Barth, M., Younglove, T., Scora, G., 2005. Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model.
3. Bektaş, T., Laporte, G., 2011. The Pollution-Routing Problem. Transp. Res. Part B Methodol. 45, 1232–1250.
4. BESTUFS, 2006. Quantification of Urban Freight Transport Effects I.
5. Demir, E., Bektaş, T., Laporte, G., 2011. A comparative analysis of several vehicle emission models for road freight transportation. Transp. Res. Part D Transp. Environ. 16, 347–357.
6. Kara, I., Kara, B.Y., Yetis, M.K., 2007. Energy Minimizing Vehicle Routing Problem, in: Dress, A., and Xu, Y., and Zhu, B. (Eds.), Combinatorial Optimization and Applications: First International Conference, COCOA. Springer Berlin Heidelberg, pp. 62–71.
7. Kirkpatrick, S., Gelatt, C.. D., Vecchi, M.. P., 1983. Optimization by Simulated Annealing. Science. 220, 671–680.
8. J. K. Lenstra, A.H.G. Rinnooy Kan., 1981. Complexity of vehicle routing and scheduling problems. Networks. 11, 221-227.
9. Li, X., Tian, P., Leung, S.C.H.H., 2010. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. Int. J. Prod. Econ. 125, 137–145.
10. Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Comput. Oper. Res. 23, 229–235.
11. Sahin, B., Yilmaz, H., Ust, Y., Guneri, A.F., Gulsun, B., 2009. An approach for analysing transportation costs and a case study. Eur. J. Oper. Res. 193, 1–11.
12. U.S. Climate Action Report 2014, 2014.

APPENDIX

Installation notes to setup the server

First, the following software need to be installed:

1. Python 3 (install Anaconda)
2. Flask for python (conda install -c anaconda flask)
3. Install mysql python link (pip install pymysql)
4. Install XAMPP

Once you have the above four installed, follow the below steps:

1. Setup MySQL in XAMPP
    - 1.1. Open XAMPP shell (windows)/ terminal (linux/mac) and type following commands and press enter
    - 1.2. "mysql -u root -p" press enter twice (no password)
    - 1.3. CREATE DATABASE gvrp_db;
    - 1.4. USE gvrp_db;
    - 1.5. CREATE TABLE input_table (NodeType varchar(20) not null, Latitude float, Longitude float, Demand float);
    - 1.6. Exit
    - 1.7. Exit
2. git clone https://github.com/surendrark/VRPonWEB/
3. Extract the cloned git files
4. Form Anaconda prompt (windows) / terminal (linux/ mac) cd to extracted folder
5. Run python GVRP.py

Now, the server is up and running.

Open a browser and type localhost:5000 or 127.0.0.1:5000 to open the web application.